Rapid Prototyping Methodology For multi-DSP TI C6X Platforms

Applied to an Mpeg-2 Coding Application

J.F Nezan, O Deforges, M Raulet

UMR IETR / Insa Rennes
20, av des Buttes de Coëmes, CS 14315
35043 RENNES Cedex, France
Phone : +33/1 233 238 459 Fax : +33/0 223 238 262
Email : jnezan@insa-rennes.fr

SPAA Revue presentation

ABSTRACT :

Real time signal, image and control applications have very important time constraints, involving the use of several powerful numerical calculation units. The aim of our project is to develop a fast prototyping process dedicated to parallel architectures made of several last generation Texas Instruments TMS320C6X DSP. The methodology is based on the use of SynDEx, a CAD software developed to improve algorithm implementation onto multiprocessor architectures, finding the best matching between an algorithm and an architecture. We have developed a SynDEx executive kernel for the C6X DSP family, in order to automatically generate a distributed and optimised static executive of the specified algorithm onto those processors. We have tested the efficiency of our methodology with a complete Mpeg-2 coding application.

1. Introduction

The aim of this work is to develop a fast prototyping process dedicated to multi-C6x architectures for image applications. We describe first our methodology which enables us to implement a complete digital signal or image processing line onto parallel architectures without any implementation pre-requirements. Next, we developed a SynDEx kernel for automatic C6x code generation, allowing an easy projection of algorithms onto multi-C6x platforms. We demonstrate finally the efficiency of our methodology with the implementation of an Mpeg-2 coding application.

2. Rapid prototyping methodology

The starting point of the prototyping process [1] is the functional description of the application, with the use of AVS [2], which is a visual development tool. AVS employs an innovative object-oriented visual programming interface to create, modify, and connect application components.

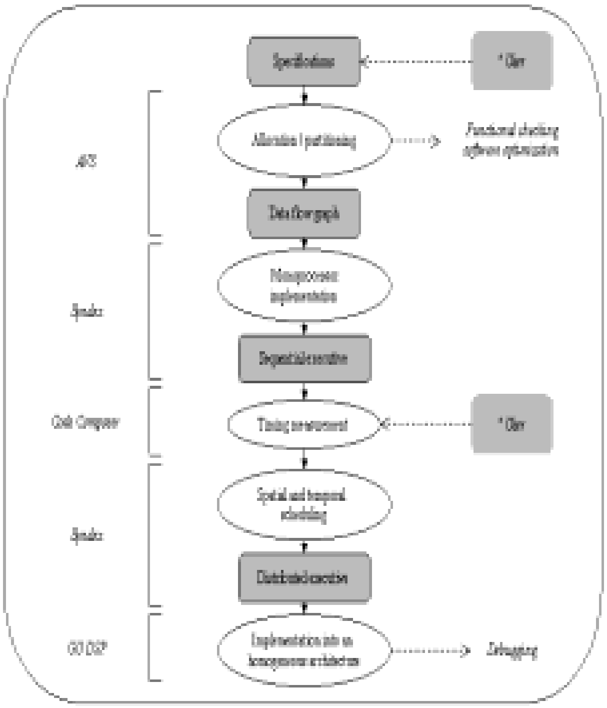

Fig 1 : Automatic Methodology for parallel architectures rapid prototyping

The description created with AVS is then automatically translated into an input file for SynDEx. SynDEx [3] is a system CAD software, supporting the AAA methodology (Adequation Algorithme Architecture) for distributed processing, which has been developed in INRIA Rocquencourt, France. The goal of adequation, (French word meaning an efficient matching) is to find the best matching between an algorithm and an architecture. Using a material graph that represents the multiprocessor architecture, and a software graph developed with AVS, SynDEx carries out the placement, the partitioning and generates an intermediate macrocode, and then the appropriated executive for the target architecture.

Except the AVS starting description, the timing measurement is the only intervention of the user (fig 1) : this step consists in determining the time associated with each function. A method to do that is to accomplish the chronometrical reports with a monoprocessor implementation, by means of the C6x debugging tool (Code Composer). The user can easily copy out these times into the software SynDEx graph. Then, SynDEx distributes all the steps among different processors.

SynDEx is able to handle different processors : Analog Device ADSP 21060, SHARC, Motorola MPC 555 et MC 68332, Intel i80x86 et i8096, Unix/Linux workstations, and Texas Instruments TMS320C40. This latter is no longer efficient enough for new applications. Therefore, our aim is to couple SynDEx advantages and C6x DSP performances creating its SynDEx automatic code generator.

3. Developed SynDEx kernel for automatic C6X code generation

From 150 to 600 Mhz clock rates, C6X are using a VLIW architecture, in order to supply up to eight 32-bit instructions to the eight functional units every clock cycle. C6x are high-performance DSPs, but not dedicated to multiprocessor architectures. Therefore, manufacturers must insert additional digital resources between two C6x in order to make their communication possible (fig 2). Inter-C6x communications are then architecture dependant.

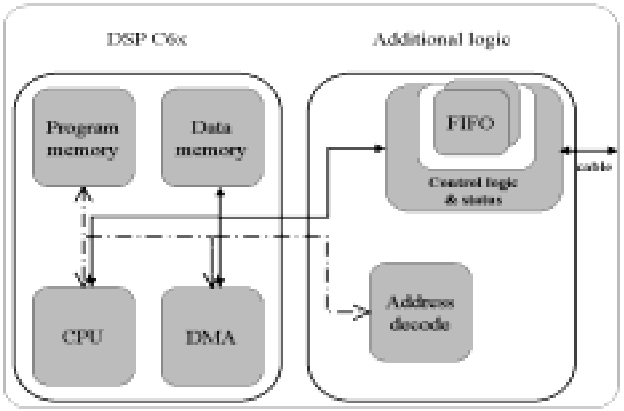

Fig 2 : C6x additional logic for communications

The executive generated by SynDEx is divided into several source files, each of them contains an intermediate code composed of a list of macro-calls of the intermediate generic kernel SynDEx. Those macro-calls are then translated by the macro-processor M4 [6] into a source code in the compilable language for target processor. We have created M4 libraries for C6x forming the SynDEx C6x Kernel.

Code Composer Studio software [4] accepts C source code and produces assembly language source code. Thanks to its optimiser, the programmer can reach high-performances without using assembly language. We decided to develop C6x Kernel in C language, in order to make it partly reusable for any C programmable DSP, and because it is not an important waste of time on a complete application.

The C6x executive kernel has been divided into several libraries, enabling its easy adaptation to a new architecture. A non-generic library contains M4 macros for the application, such as specific input/output functions. Architecture independent library contains macros used whatever the architecture target. The other libraries connected with processor type, processor or communication type dependent are architecture dependent. We developed kernels for two different platforms to ensure that macros can be reused. The adaptation of our work for another multi-C6x architecture is limited to the communication sequence adaptation for a new media, if needed.

SynDEx macrocode creates two interleaved schedulers : one for computation tasks and the other for communications, allowing parallelism of those actions. We have chosen the use of multi-channel DMA transfers (fig 3), maximizing this parallelism and timing performances. The DMA contains four different channels, we have linked each of them to one SynDEx communication media. By this way, SynDEx architecture graph can have four connections for each C6x.

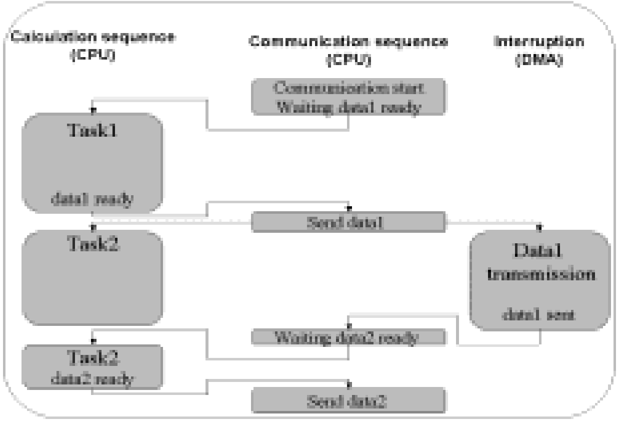

Fig 3 : parallelism computation/communication

4. Mpeg-2 application results

We developed AVS modules for an Mpeg-2 coding of rectangular video documents at main profile and main level (main@main). The average time for the sequential coding of a single 128*128 image with our algorithm is valued at 910 ms over a single C40 DSP, if all data could be put in internal memories. The resulting time spent to code an image should be 620 ms using our methodology with a three C40 platform, that is to say 1.47 speed up factor.

We implemented the same algorithm over a single C6201 DSP. Chronometrical reports, done with C6x timers, give 331 ms for the whole coding process. The code used for our tests is from TM5 software given with the Mpeg-2 standard. Resources are not optimised for DSP prototyping. So, many data must be placed on external memories, slowing down calculations [5].

Thanks to the fast communication links allowed for C6x architectures, the partitioning of the application is more efficient : the mpeg-2 coding over a two C6201 platform reaches 182 ms, a 1.81 speed up factor.

We can notice that, with the methodology used, the application which was developed with AVS for multi-C40 architectures is directly usable for multi-C6x platforms.

5. Conclusion :

We have created an automatic distributed executive generator for multi-C6x DSP architectures using SynDEx. It was tested onto Texas Instruments TMS320C6201, but it is suitable for other C6x family's DSPs. Furthermore, because of the use of C language, it can serve as a basis for other DSP kernel developments. Static executives generated are custom-built, avoiding to add operating systems and saving platform resources.

We are currently developing new Mpeg-4 AVS modules, optimising C functions for DSP implementations. By this way, most of resources of the algorithms will be placed on internal memories. The result should be real-time Mpeg applications.

We are also working with SynDEx V6 developers on the automatic code generator with new features : shared memory, conditional nodes and hierarchy in the SynDEx algorithm graph description. The last issue will allow the description and implementation of application with partial or full reconfigurability.

The additional logic added between two C6x DSPs is often integrated in a FPGA. The implementation of elementary and regular operations onto this material part would give higher performances. Therefore, we plan to study the adding of this material element in the SynDEx material graph, and the generation code for FPGA.

6. References

[1] V. Fresse, M. Assouil, O. Deforges : *Rapid prototyping of image processing onto a multiprocessor architecture,* DSP World ICSPAT, Orlando, Florida, USA, Nov. 1999.

[2] International AVS Center, Manchester Visualization Centre, Manchester Computing, University of Manchester. Available at http://www.iavsc.org.

[3] T.Grandpierre, C.Lavarenne, Y.Sorel : *optimized rapid prototyping for real time embedded heterogeneous multiprocessors*. In 7th International workshop on Hardware/Software Co-Design. IEEE Computer Society, ACM SIGSOFT, IFIP. Rome, Italy, May 99.

[4] Texas Instruments technical documents : *TMS320C6000 Code Composer Studio - tutorial.* Ref spru301c. Available at http://www.dspvillage.ti.com. Feb 00.

[5] Hunt Engineering technical document : *external memory types for C6000 Systems.* Available at http://www. Hunteng.co.uk. May 99.

[6] M. Loukides, A. Oram : *Programming with GNU software.*